



# Population Markov Chain Monte Carlo and Genetic Networks

Fujun Ye

MSc in Artificial Intelligence  
Supervised by Dirk Husmeier

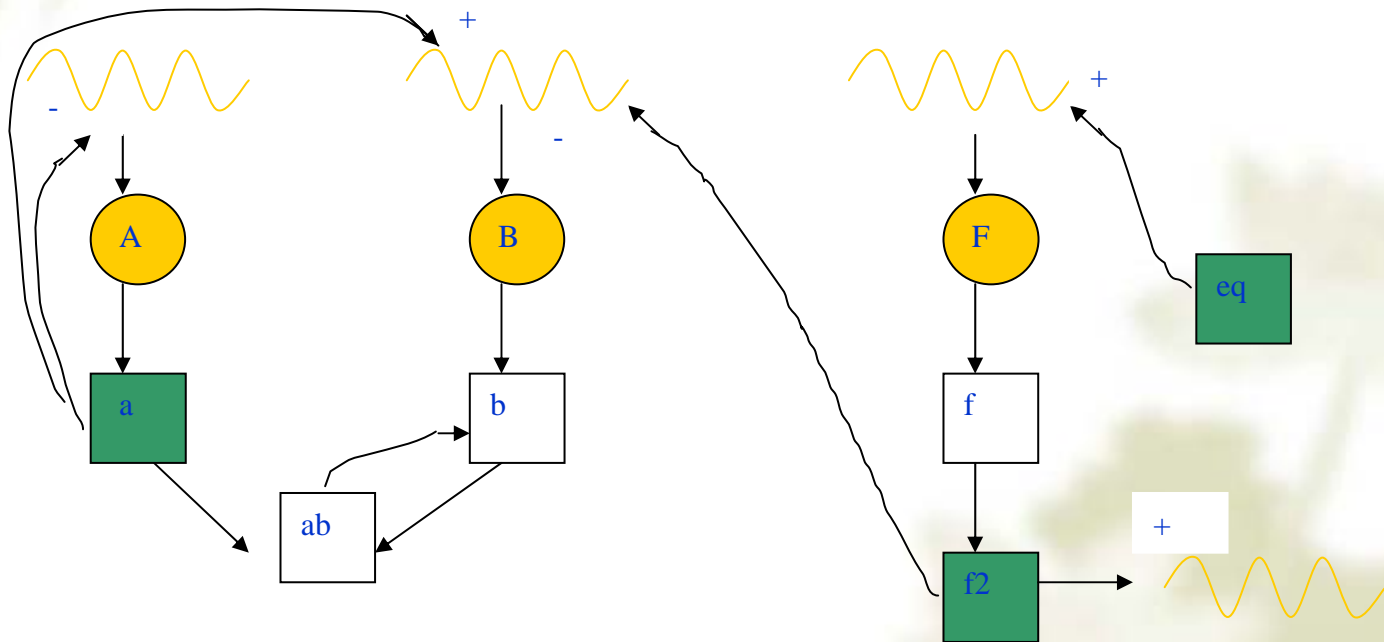
# Outline

- ❖ Introduction
- ❖ MCMCMC
- ❖ MCMCMC for missing values
- ❖ Result Evaluation (complete data)
- ❖ Result Evaluation (missing values)
- ❖ Summary

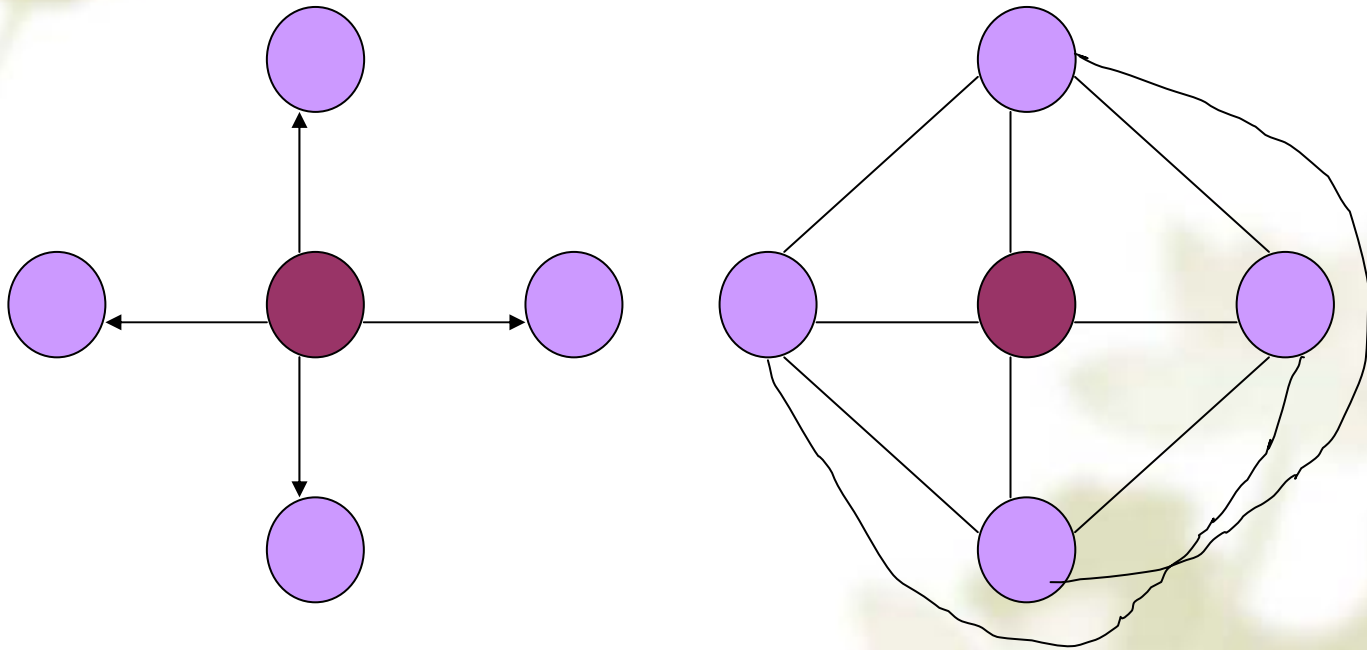
# Introduction

- ❖ Genetic Network
- ❖ Clustering and Differential equation
- ❖ Bayesian Network
- ❖ MCMC

# Genetic Network



# Clustering



# Differential Equation

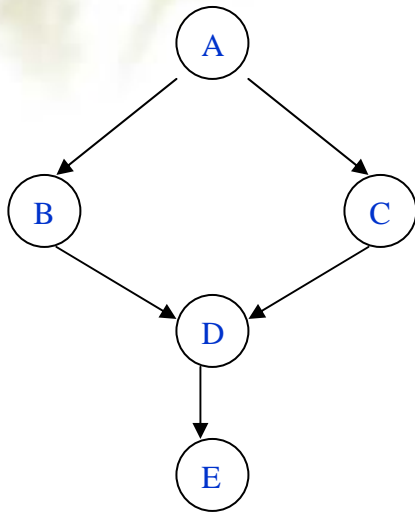
## ❖ Advantage

provide detailed understanding of the biological systems

## ❖ Shortcoming

short of data  
noisy data

# Inferring Bayesian Network From Expression Data



## ❖ Bayesian Network

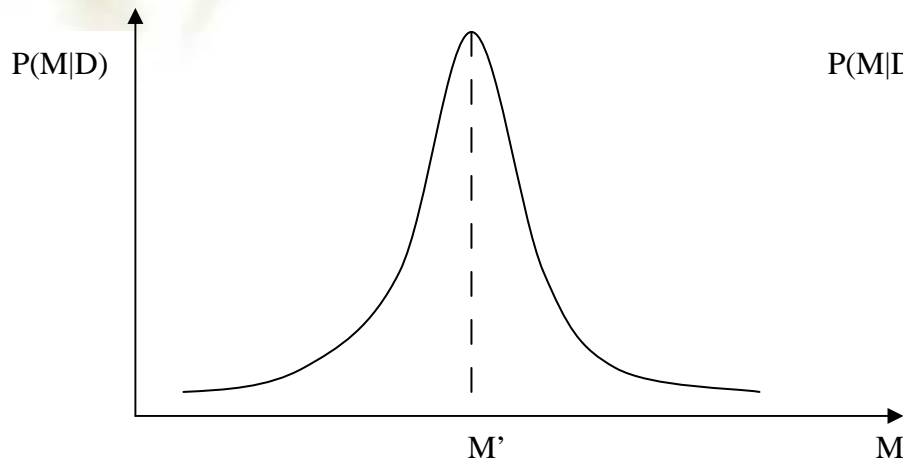
$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i | Pa_G(X_i))$$

$$P(a, b, c, d, e) = P(e | d)P(d | b, c)P(c | a)P(b | a)P(a)$$

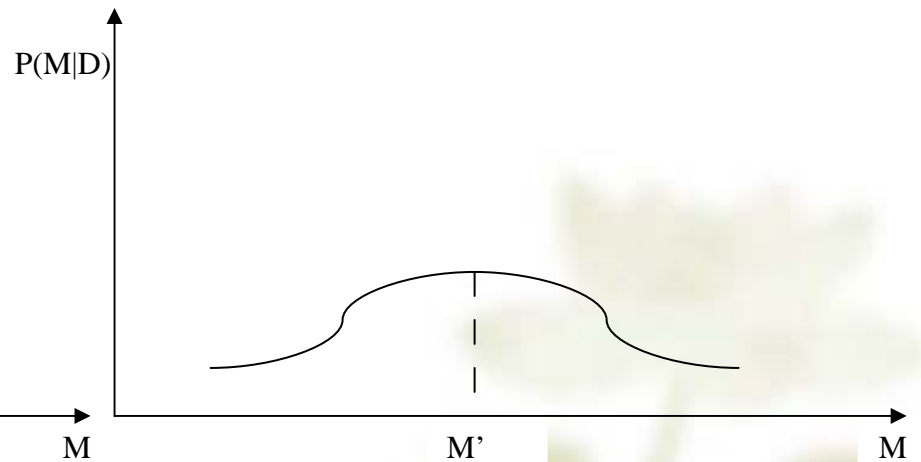
# Problems

- ❖ The number of different network structures grows super exponentially with the number of nodes

|                 |   |     |                   |                      |                      |
|-----------------|---|-----|-------------------|----------------------|----------------------|
| N of nodes      | 2 | 4   | 6                 | 8                    | 10                   |
| N of structures | 3 | 543 | $3.7 \times 10^6$ | $7.8 \times 10^{11}$ | $4.2 \times 10^{18}$ |



Where the data set is large, the optimal structure  $M'$  is well defined



Where the data set is small, there are many networks which can explain the data fairly well.

# MCMC

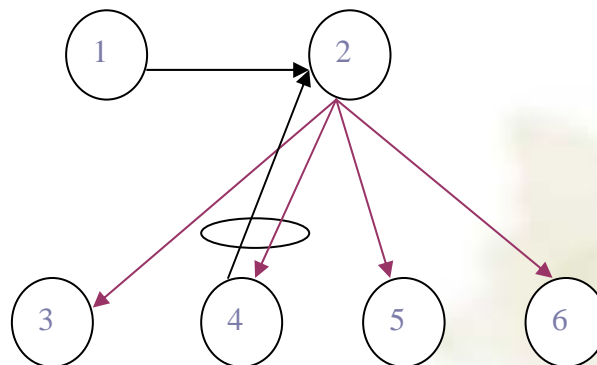
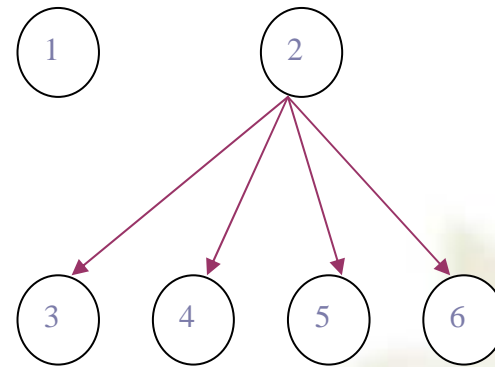
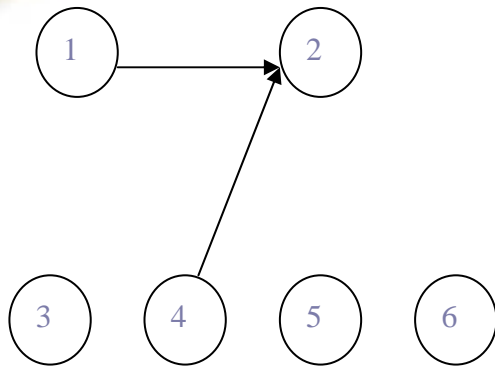
- ❖ MCMC samples networks from its posterior distribution

$$P(M_k | D) = \frac{P(D | M_k)P(M_k)}{\sum_i P(D | M_i)P(M_i)}$$

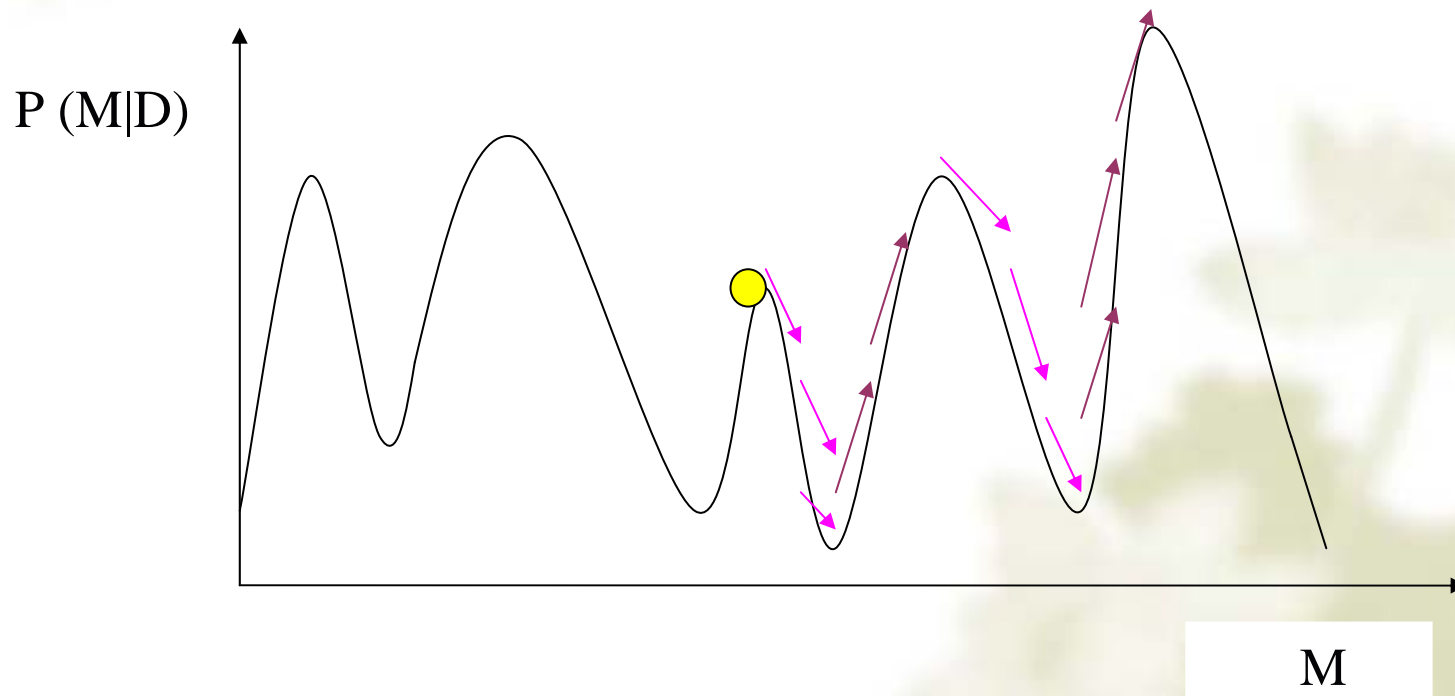
- ❖ Calculate the posterior probability of a feature

$$P(f | D) = \frac{\sum_i P(M_i | D)P(f | M_i)}{\sum_i P(M_i | D)}$$

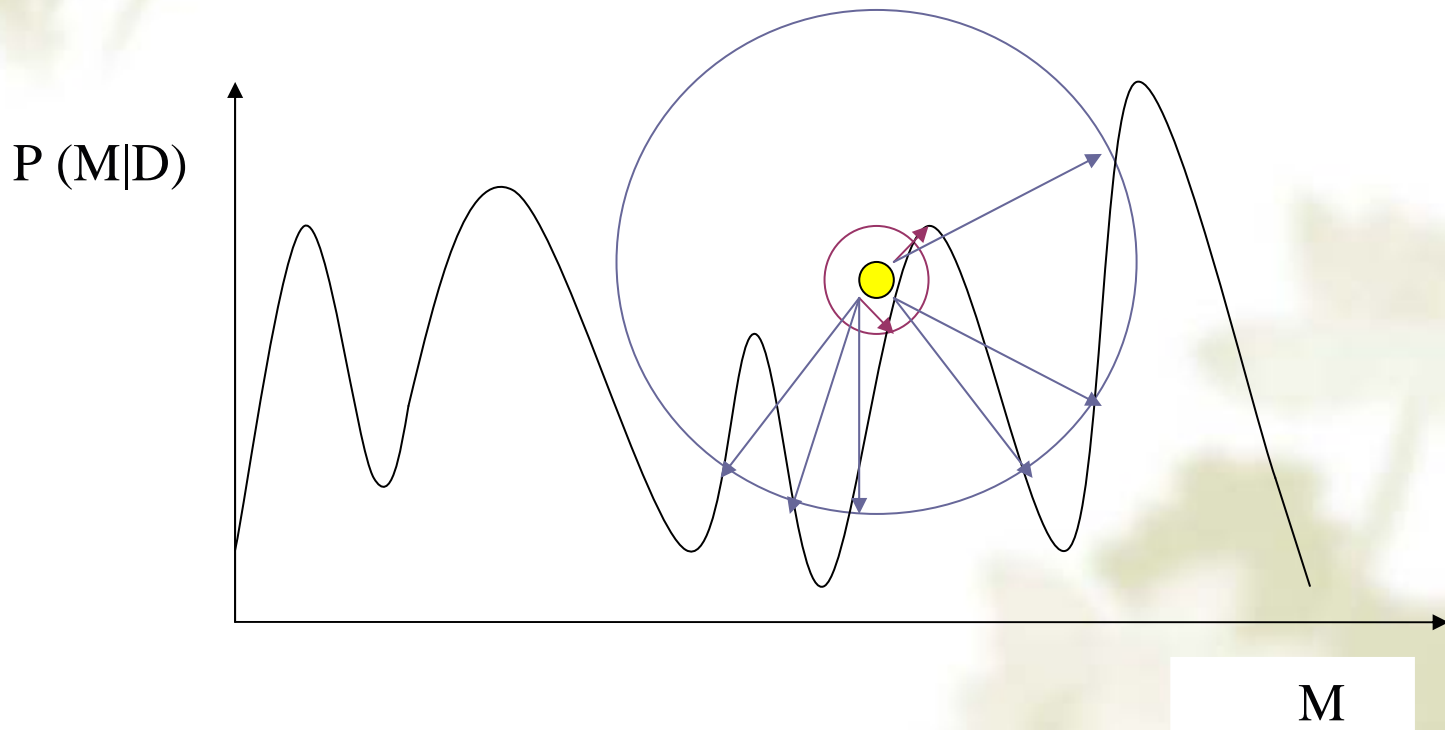
## ❖ Coincidence dependence



❖ Escape from local optima using traditional MCMC



❖ Small step size versus big step size



# Problems

- ❖ Huge search space and coincidence dependence — Prescreening is important!
- ❖ Local optima — Traversal operator is important!
- ❖ Fixed step size — Varied step size is more reasonable

# MCMCMC

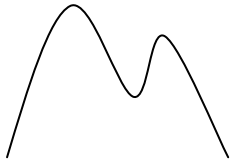
- ❖ Metropolis-coupled Markov Chain Monte Carlo (MCMCMC)
- ❖ Pre-processing method
- ❖ Traversal operators
- ❖ Algorithm
- ❖ MCMCMC for missing values

# MCMCMC

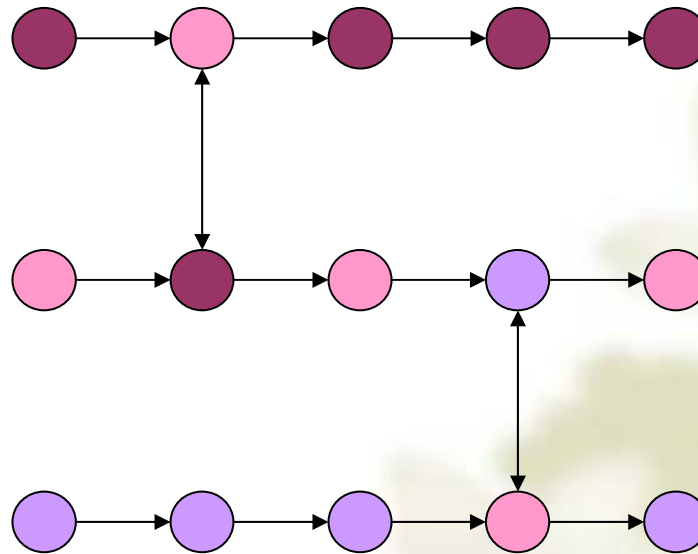
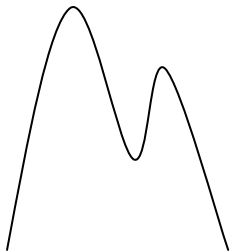
$$T_3 > T_2$$



$$T_2 > T_1$$



$$T_1 = 1$$



- ❖ For each chain, move a step based on

$$A(M', M) = \min\left(1, \left(\frac{P(D | M')P(M')}{P(D | M)P(M)}\right)^{1/T} \frac{Q(M | M')}{Q(M' | M)}\right)$$

- ❖ Chain swap

$$S_a = \begin{pmatrix} T_1 & T_2 & \dots & T_i & \dots & T_k & \dots & T_M \\ M_1 & M_2 & \dots & M_i & \dots & M_k & \dots & M_M \end{pmatrix}$$

$$S_b = \begin{pmatrix} T_1 & T_2 & \dots & T_i & \dots & T_k & \dots & T_M \\ M_1 & M_2 & \dots & M_k & \dots & M_i & \dots & M_M \end{pmatrix}$$

**Acceptance Probability**

$$P_a = \min\left\{1, \frac{[P(D | M_i)P(M_i)]^{1/T_k} [P(D | M_k)P(M_k)]^{1/T_i}}{[P(D | M_i)P(M_i)]^{1/T_i} [P(D | M_k)P(M_k)]^{1/T_k}}\right\}$$

# Pre-processing method

$$P(D | M) = \int P(D | \theta, M) P(\theta | M) d\theta$$

❖ Penalize complex model

$$\alpha_{nv_n\pi_n} = \frac{\alpha}{|v_n| * |\pi_n|} \quad \sum_{v_n} \alpha_{nv_v\pi_n} = \alpha_{n\pi_n}$$

$$p(D | M) = \prod_n \prod_{\pi_n} \frac{\Gamma(\alpha_{n\pi_n})}{\Gamma(\alpha_{n\pi_n} + n_{n\pi_n})} \prod_{v_n} \frac{\Gamma(\alpha_{nv_n\pi_n} + n_{nv_n\pi_n})}{\Gamma(\alpha_{nv_n\pi_n})}$$

The log likelihood is

$$\log(p(D | M)) = \sum_n \text{score}(n, \pi_n, D)$$

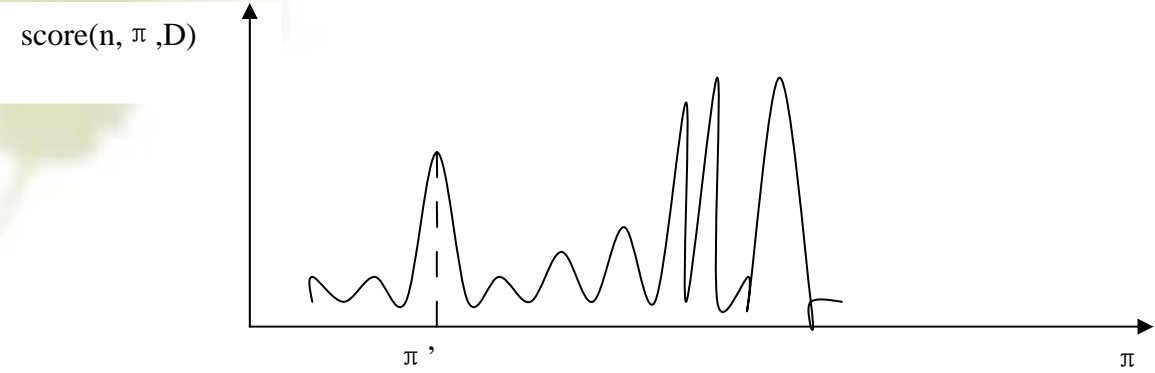
where

$$\begin{aligned} \text{score}(n, \pi_n, D) = & \sum_{\pi_n} [\log(\Gamma(\alpha_{n\pi_n})) - \log(\Gamma(\alpha_{nv_n\pi_n} + n_{n\pi_n}))] + \\ & \sum_{\pi_n} \sum_{v_n} [\log(\Gamma(\alpha_{nv_n\pi_n} + n_{nv_n\pi_n})) - \log(\Gamma(\alpha_{nv_n\pi_n}))] \end{aligned}$$

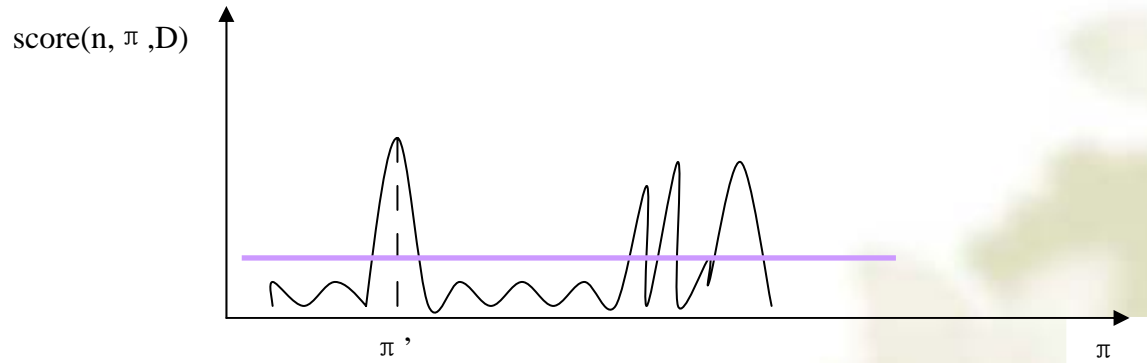
- ❖ Use some max fan in
- ❖ Find all possible parents-configurations for each node and delete low score parents-configurations
- ❖ Keep C parents-configurations for each node and cardinality

Threshold is set as:

$$\theta = \lambda * (score_{sh} - score_{sl}) / m + score_{sl} / m$$



When data is quite sparse and noisy



Using pre-screening method

# Traversal operators

## ❖ Importance sampling---

Sample a parents-configuration for a node

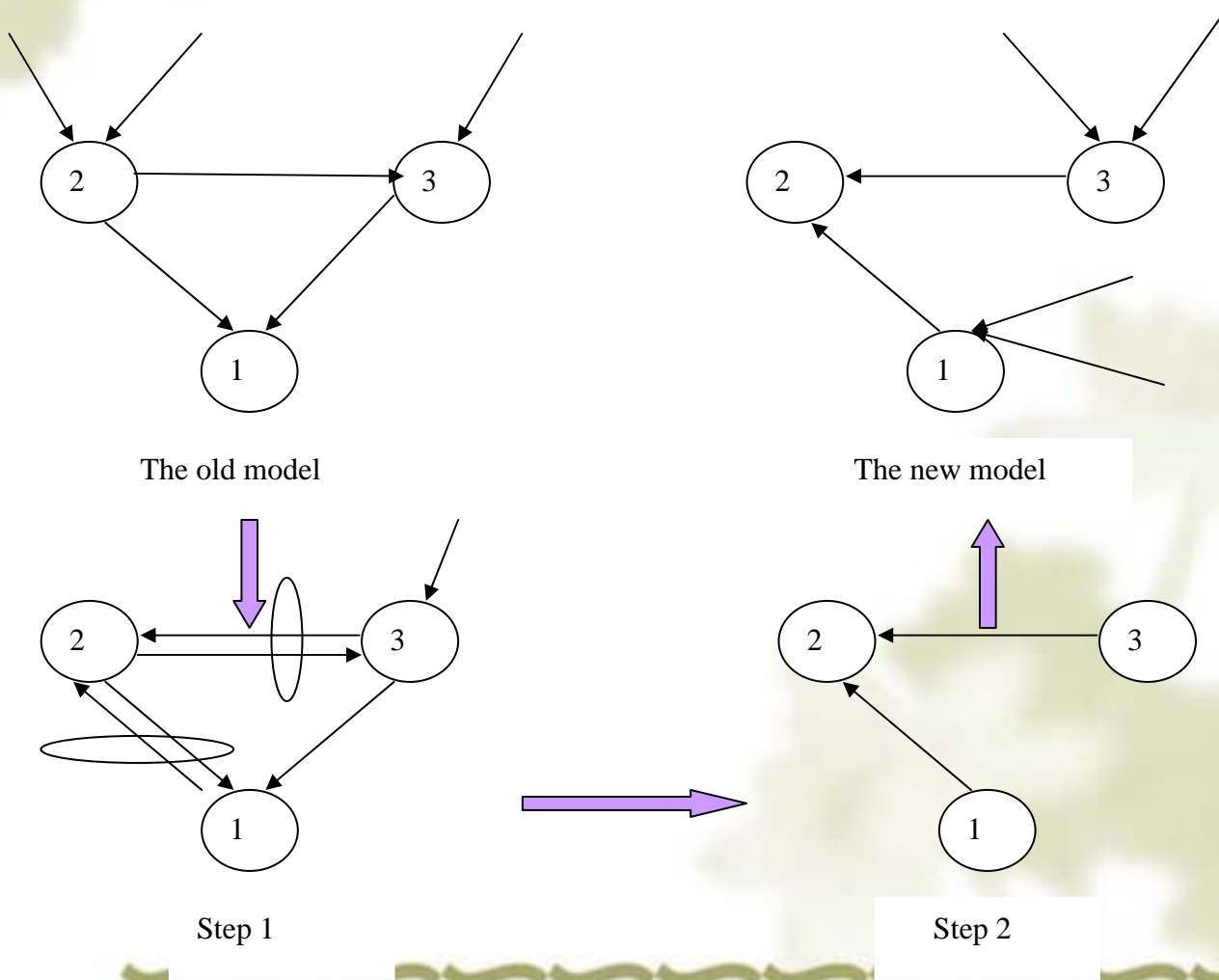
$$P(\text{node } i) = \frac{K_i}{\sum_{i=1}^n K_i} \quad p(\pi_j | \text{node } i) = \frac{\text{score}(i, \pi_j, D) + C}{\sum_{k=1, k \neq k_{\text{old}}}^n \text{score}(i, \pi_k, D) + (n-1)C}$$

$$\frac{Q(M_{\text{old}} | M_{\text{new}})}{Q(M_{\text{new}} | M_{\text{old}})} = \frac{\text{score}(i, \pi_{k_{\text{old}}}, D) + C}{\sum_{k=1, k \neq k_{\text{old}}}^n \text{score}(i, \pi_k, D) + (n-1)C}$$

$$\frac{\text{score}(i, \pi_{k_{\text{new}}}, D) + C}{\sum_{k=1, k \neq k_{\text{new}}}^n \text{score}(i, \pi_k, D) + (n-1)C}$$

$$\frac{P(D | M_{\text{new}})}{P(D | M_{\text{old}})} = \exp(\text{score}(i, \pi_{k_{\text{new}}}, D) - \text{score}(i, \pi_{k_{\text{old}}}, D))$$

# ❖ DIN sampling --- If the new network is loopy

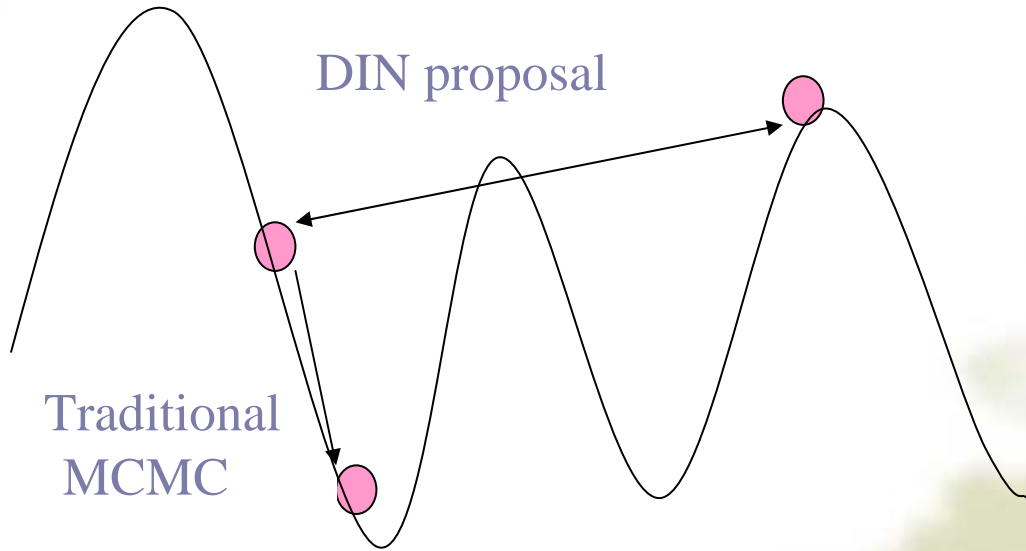


$$A(Mnew | Mold) = \text{Min}\left(1, \frac{P(D | Mnew) * P(Mnew) * Q(Mold | Mnew)}{P(D | Mold) * P(Mold) * Q(Mnew | Mold)}\right)$$

$$\frac{P(D | Mnew)}{P(D | Mold)} = \frac{\exp\left(\sum_{j=1}^n \text{score}(n_j, n\pi(n_j), D) + \text{score}(n_i, n\pi(n_i), D)\right)}{\exp\left(\sum_{j=1}^n \text{score}(n_j, o\pi(n_j), D) + \text{score}(n_i, o\pi(n_i), D)\right)}$$

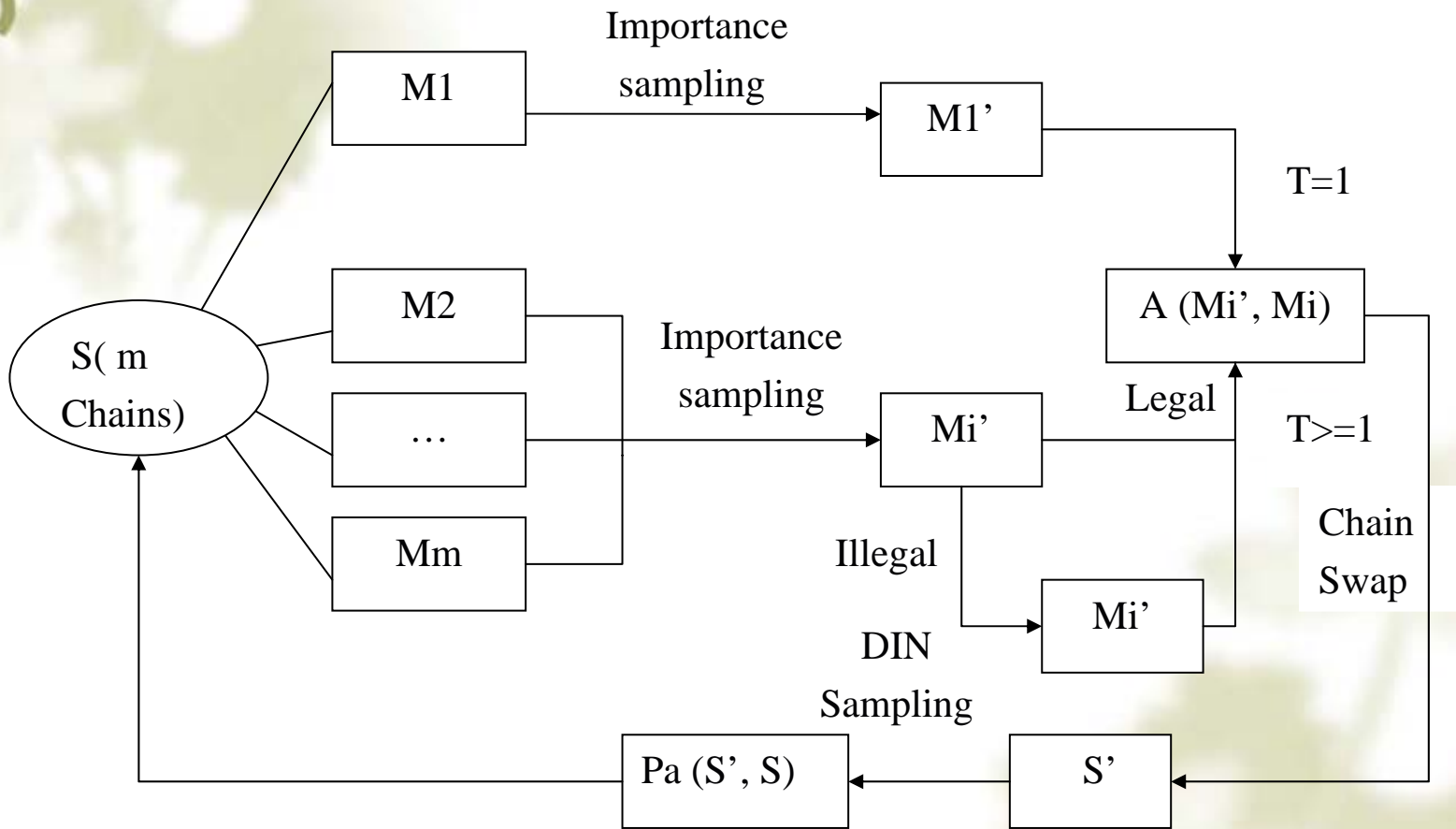
$$\frac{Q(Mold | Mnew)}{Q(Mnew | Mold)} = \frac{\text{score}(i, \pi_{k\_old}, D) + C}{\sum_{k=1, k \neq k\_old}^n \text{score}(i, \pi_k, D) + (n-1)C} \Bigg/ \frac{\text{score}(i, \pi_{k\_new}, D) + C}{\sum_{k=1, k \neq k\_new}^n \text{score}(i, \pi_k, D) + (n-1)C}$$

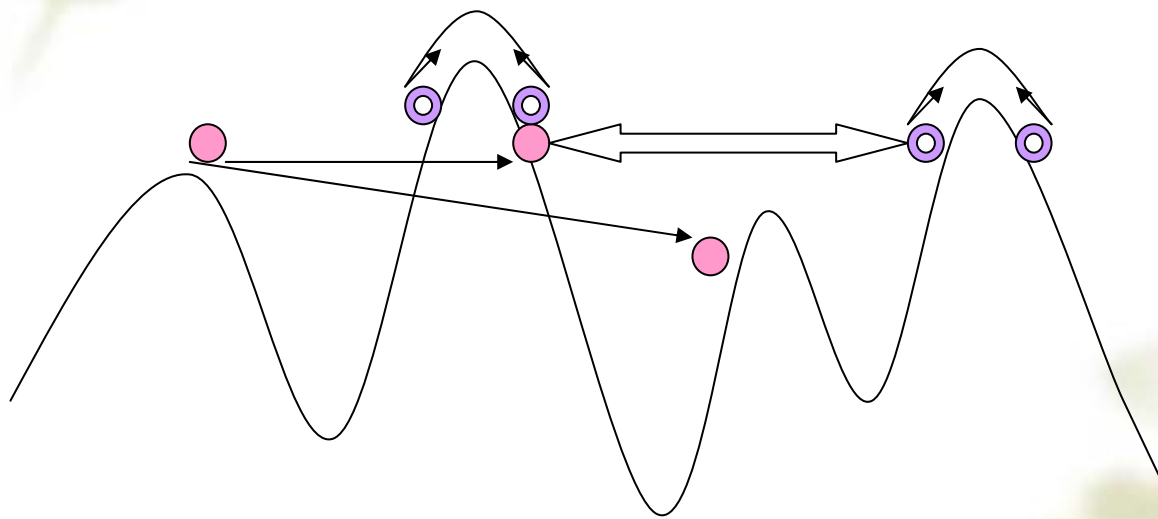
I simply use an approximation since it is quite time consuming to calculate the proposal probability



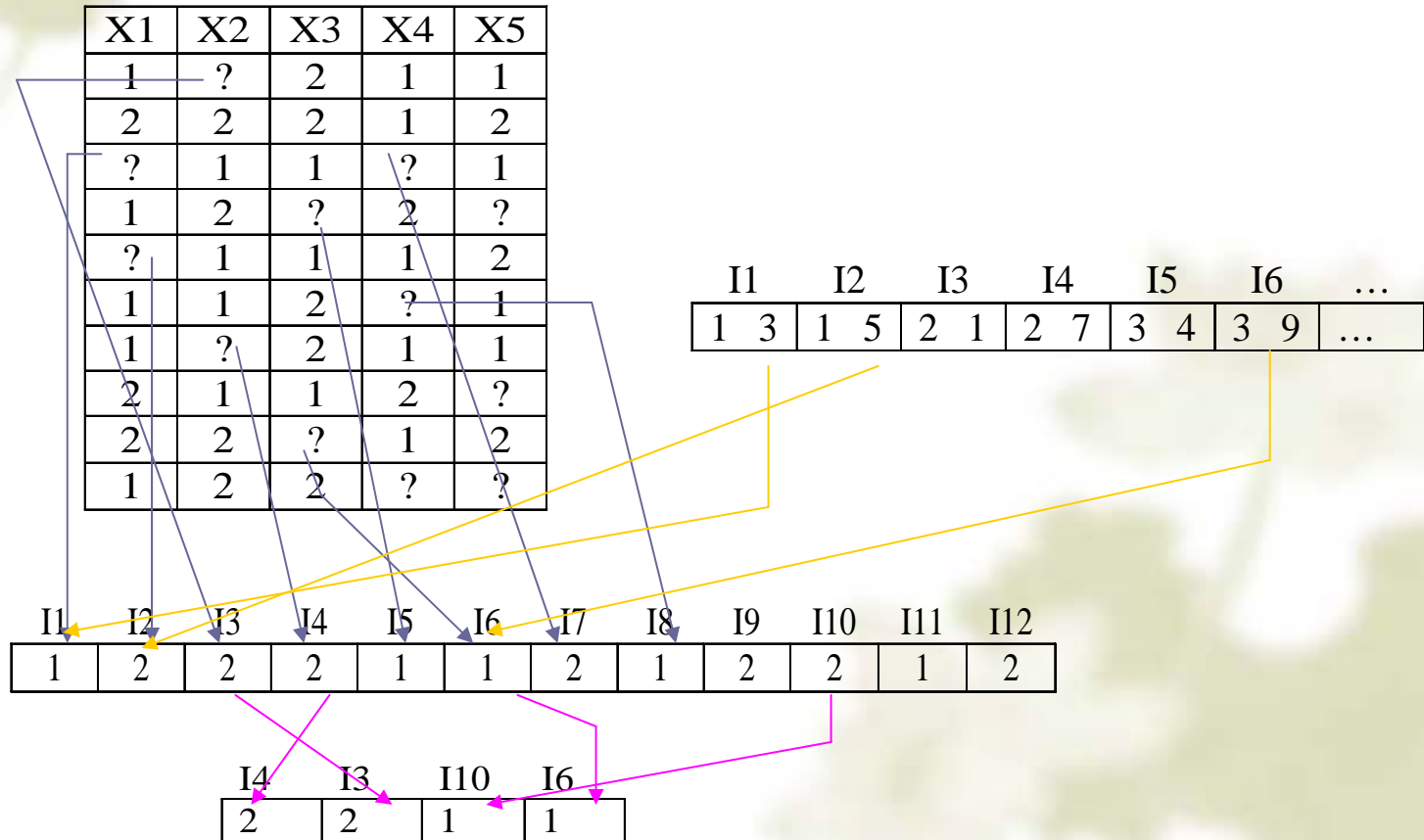
# Algorithm

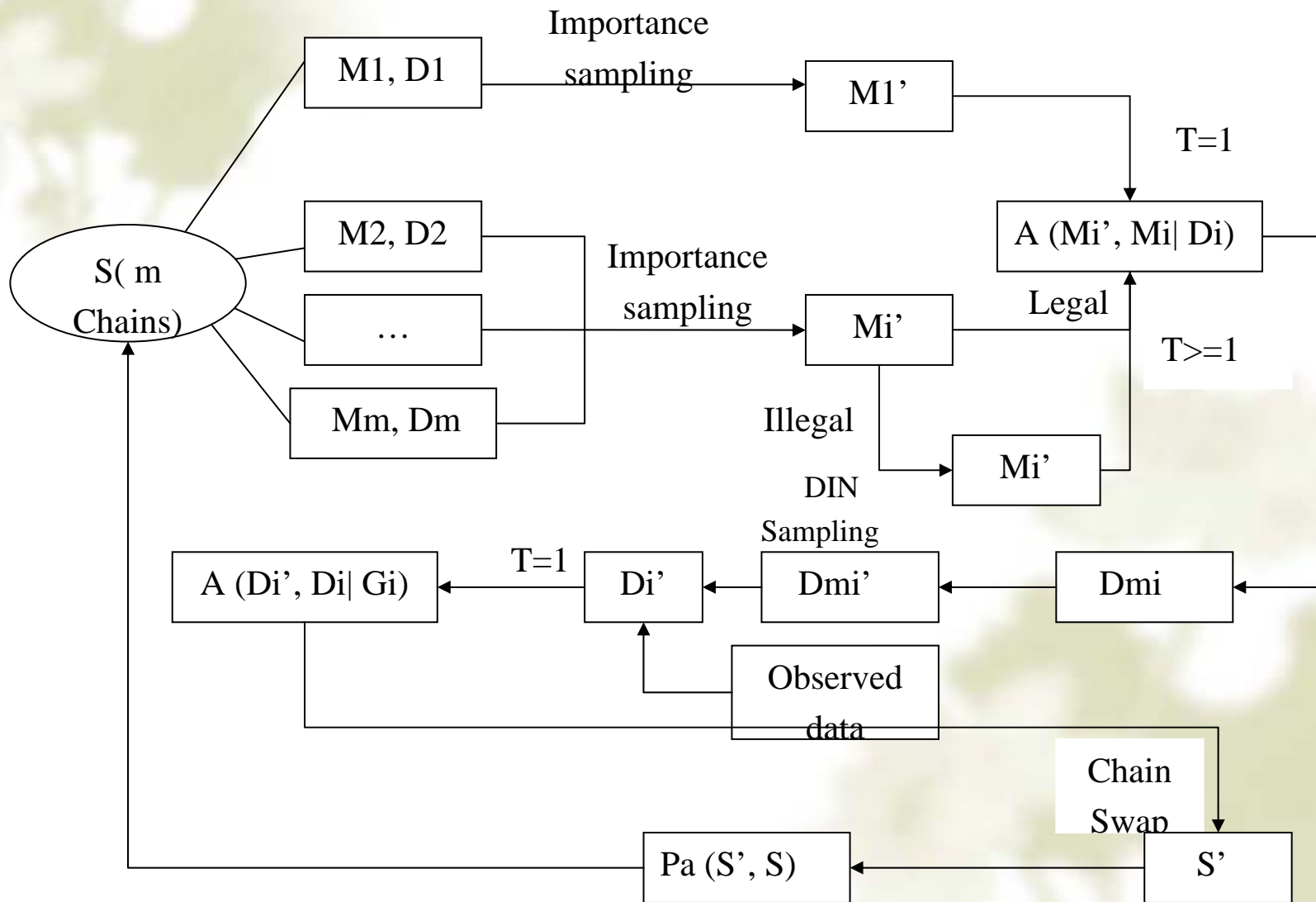
- ❖ Initialization
- ❖ Each iteration
  - Move a step for every chain
  - Chain swap
- ❖ Keep the first chain





# MCMCMC for missing values



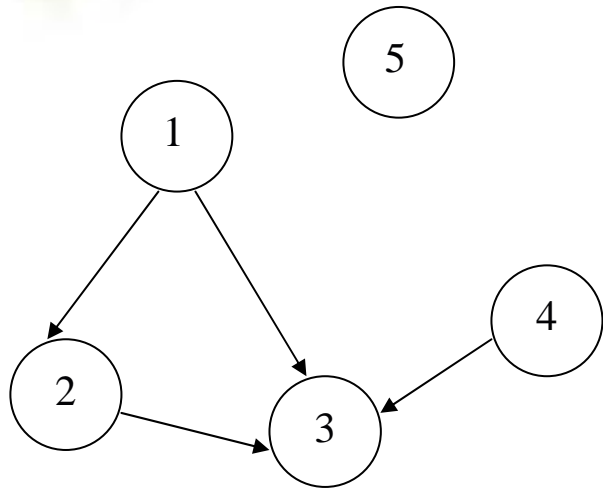


- ❖ Proposal method for before burn in

$$Q(v_n | M, n) = \frac{N_{v_n} + 1}{\sum_{v_n} (N_{v_n} + 1)}$$

$$Q(v_n | M, n, m) = \frac{N_{v_n v_{\pi n}} + 1}{\sum_{v_n} (N_{v_n v_{\pi n}} + 1)}$$

$$Q(v_n, v_{\pi mis} | M, n, m) = \frac{N_{v_n v_{\pi mis} v_{\pi m}} + 1}{\sum_{v_n, \pi mis} (N_{v_n v_{\pi mis} v_{\pi m}} + 1)}$$



| X1 | X2 | X3 | X4 | X5 |
|----|----|----|----|----|
| 1  | ?  | 2  | 1  | 1  |
| 2  | 2  | 2  | 1  | 2  |
| ?  | 1  | 1  | 2  | 1  |
| 1  | 2  | ?  | ?  | ?  |
| ?  | 1  | 1  | 1  | 2  |
| 1  | 1  | 2  | ?  | 1  |
| 1  | ?  | 2  | 1  | 1  |
| 2  | 1  | 1  | 2  | ?  |
| 2  | 2  | ?  | 1  | 2  |
| 1  | 2  | 2  | ?  | ?  |

## Acceptance probability

$$\text{Accept}(\text{MissVal}, \text{MissVal}') = \min\left(1, \frac{Q(\text{MissVal} \mid \text{MissVal}')P(D' \mid M)}{Q(\text{MissVal}' \mid \text{MissVal})P(D \mid M)}\right)$$

❖ After burn in

$$Q(MissVal' | MissVal) = \prod_{i \in \Omega(cmis)} \frac{N_{i\_new} + 1}{\sum_j (N_{ij} + 1)}$$

$$Q(MissVal | MissVal') = \prod_{i \in \Omega(cmis)} \frac{N_{i\_old} + 1}{\sum_j (N_{ij}' + 1)}$$

**Acceptance probability**

$$Accept(MissVal, MissVal') = \min\left(1, \frac{Q(MissVal | MissVal')P(D' | M)}{Q(MissVal' | MissVal)P(D | M)}\right)$$

# Result Evaluation (complete data)

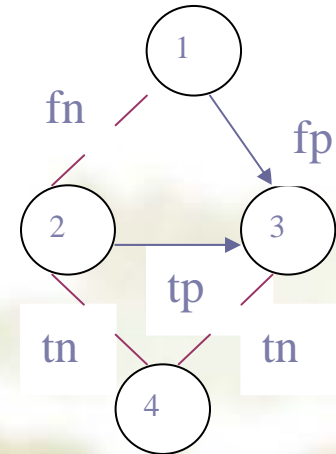
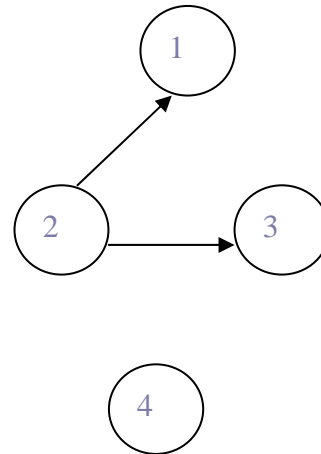
## ❖ ROC curve

$$\text{sensitivity} = \frac{tp}{tp + fn}$$

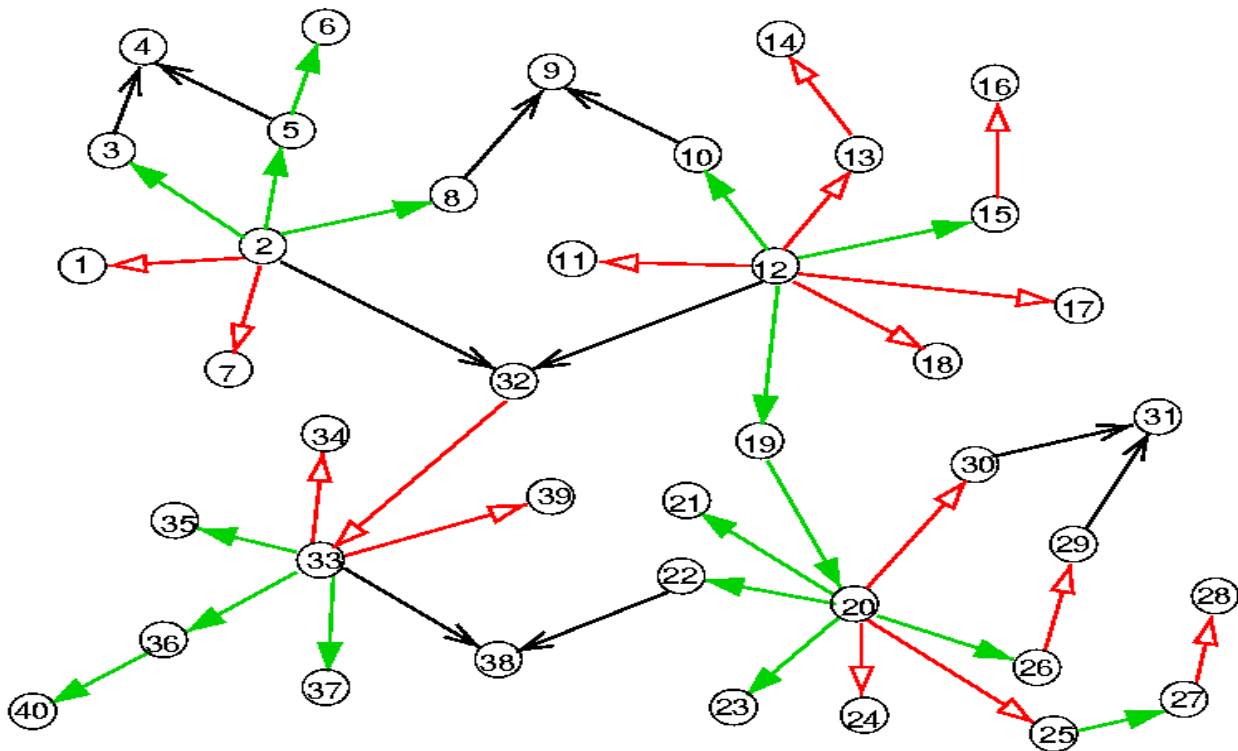
$$\text{specificity} = \frac{tn}{tn + fp}$$

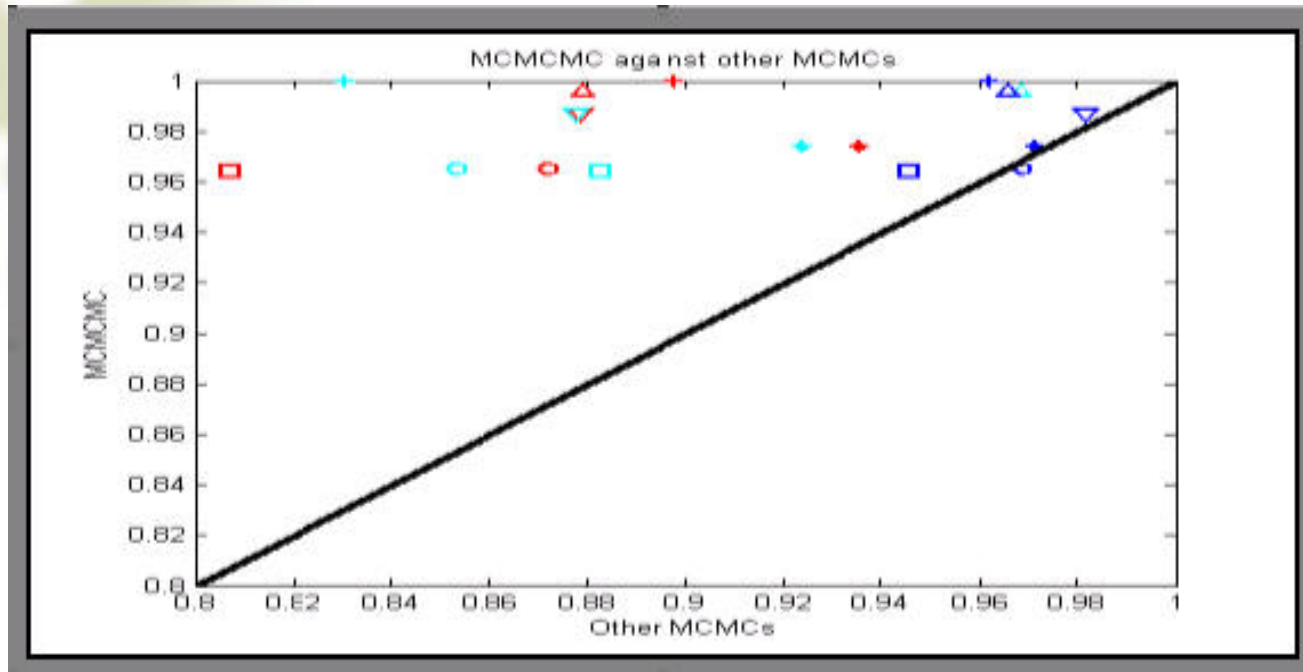
$$\text{complementary specificity} = 1 - \frac{tn}{tn + fp} = \frac{fp}{tn + fp}$$

$tp$  is the number of true positive edges.  $fn$  is the number of false negative edges.  
 $fp$  is the number of false positive edges.  $tn$  is the number of true negative edges.

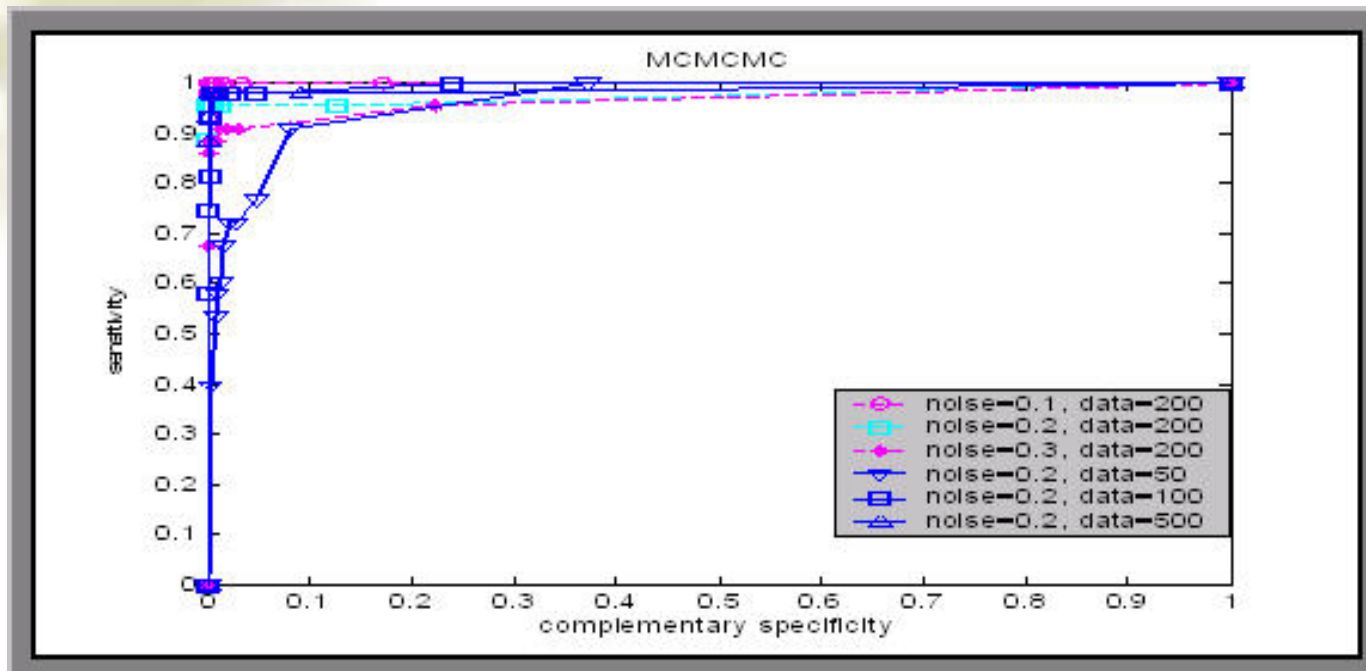


## ❖ Model Genetic Network



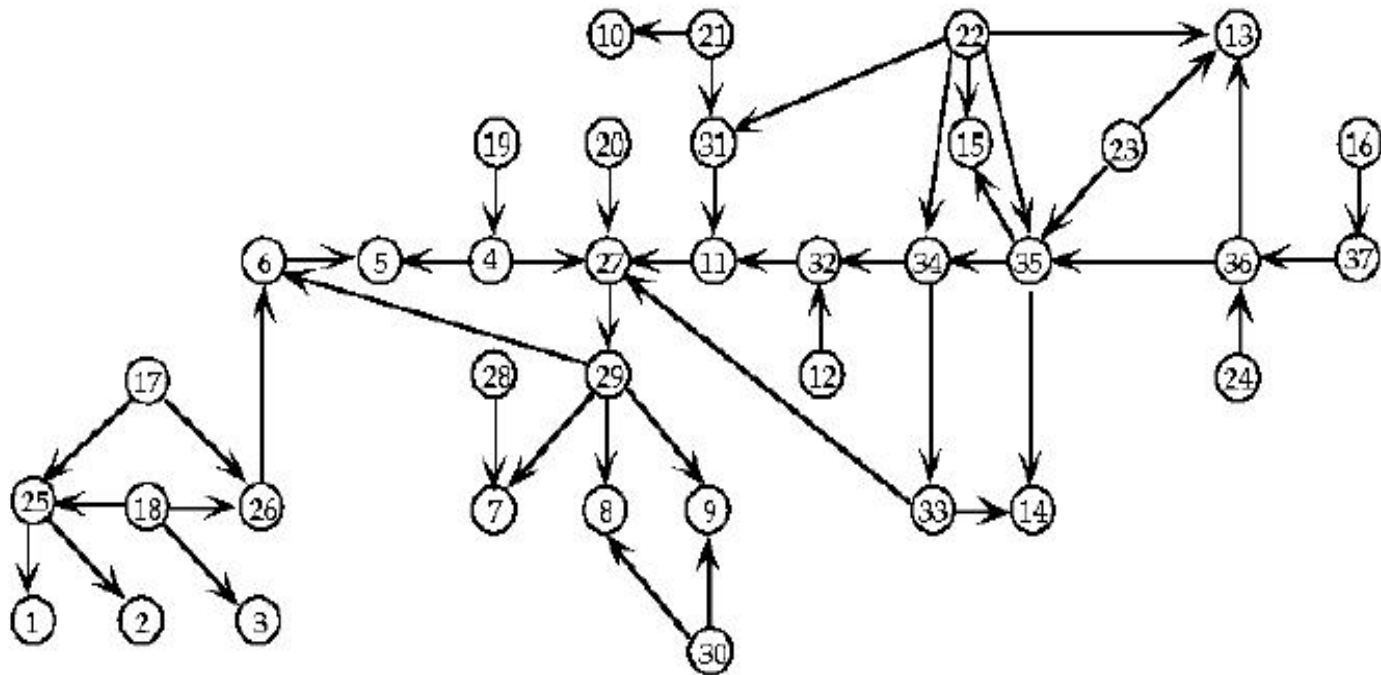


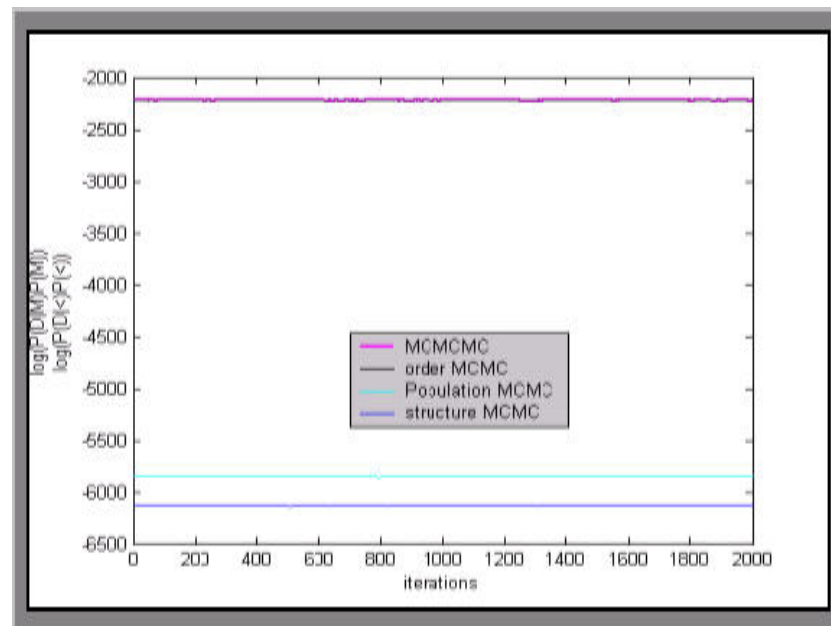
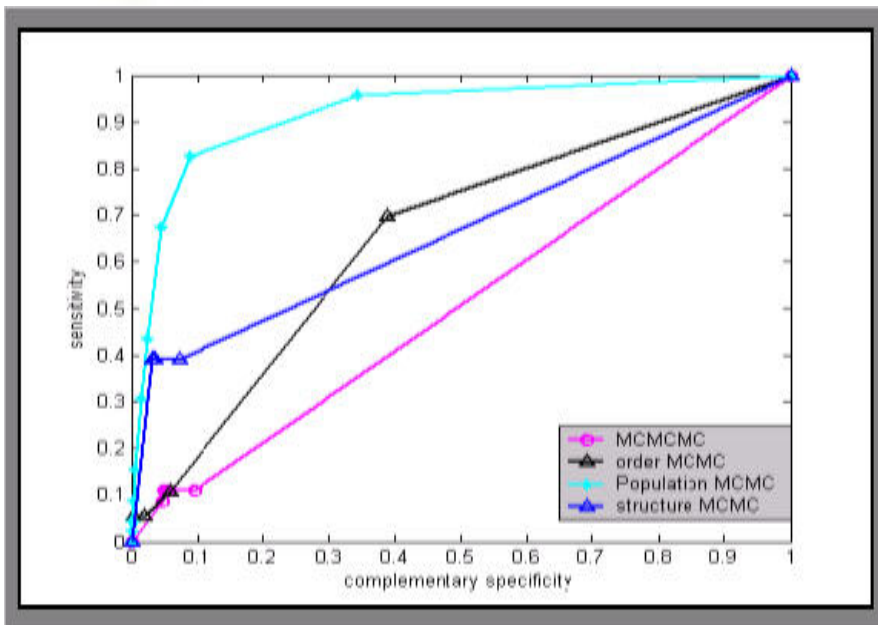
- ❖ MCMCMC against order MCMC
- ❖ MCMCMC against structure MCMC
- ❖ MCMCMC against Population MCMC



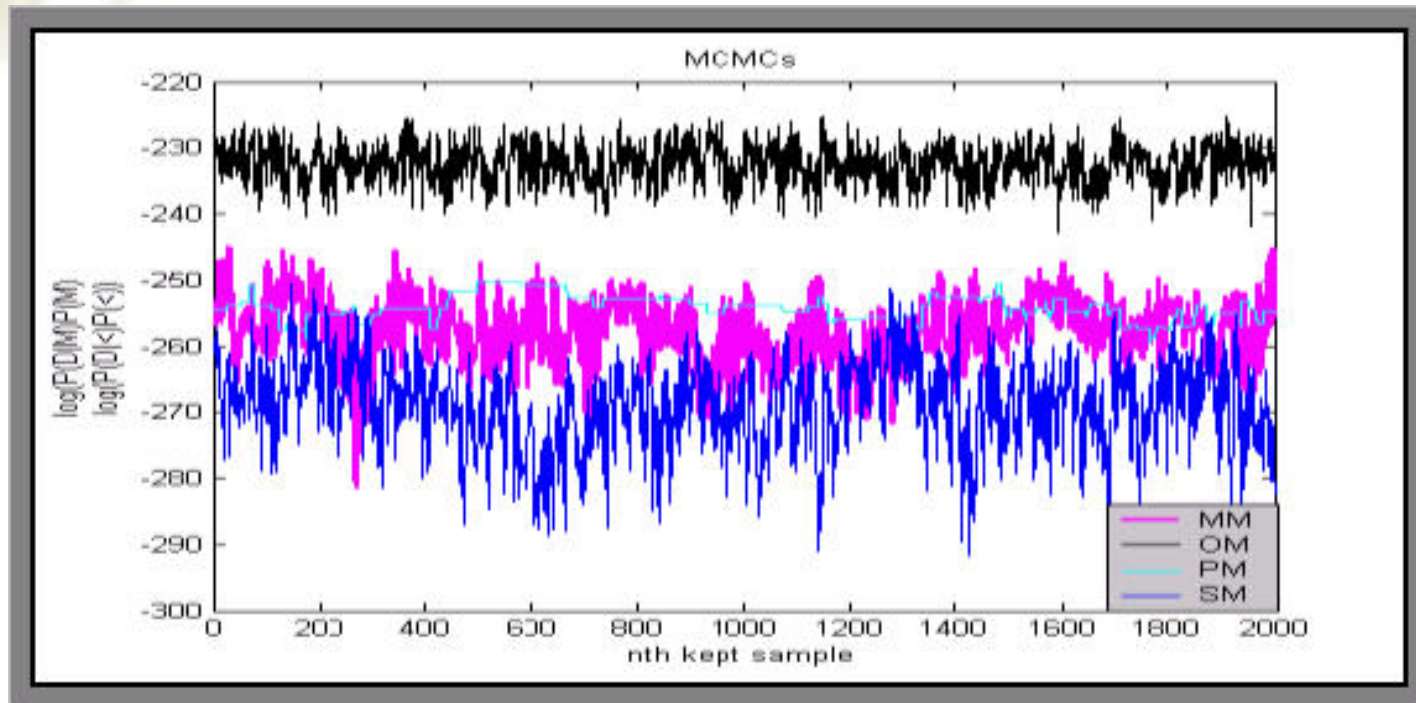
- ❖ Temperatures= [1, 1, 3, 9, 30]
- ❖ Keep at most 10 parents-configurations for each node and cardinality.
- ❖ With 60000 iterations: 30000 burn in and keep the last 30000 samples.

## ❖ Alarm Network



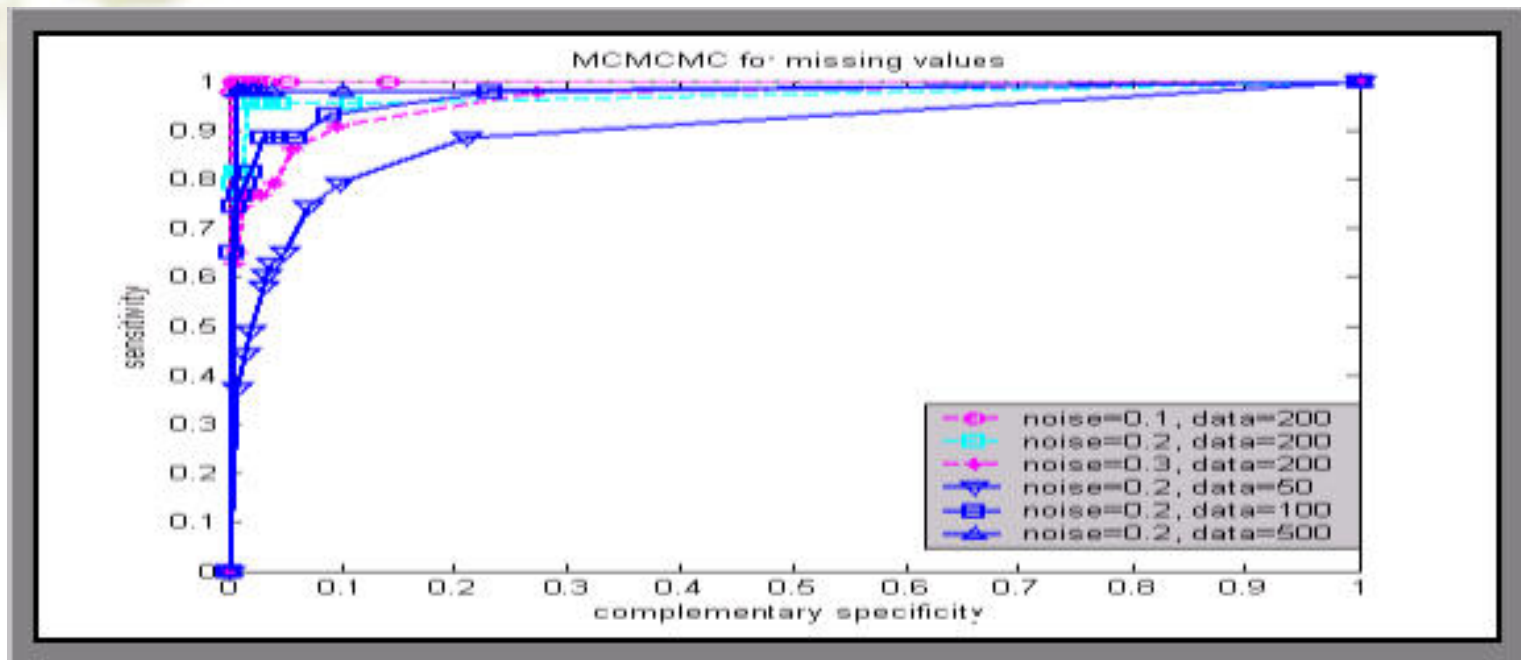


## ❖ Arabidopsis data

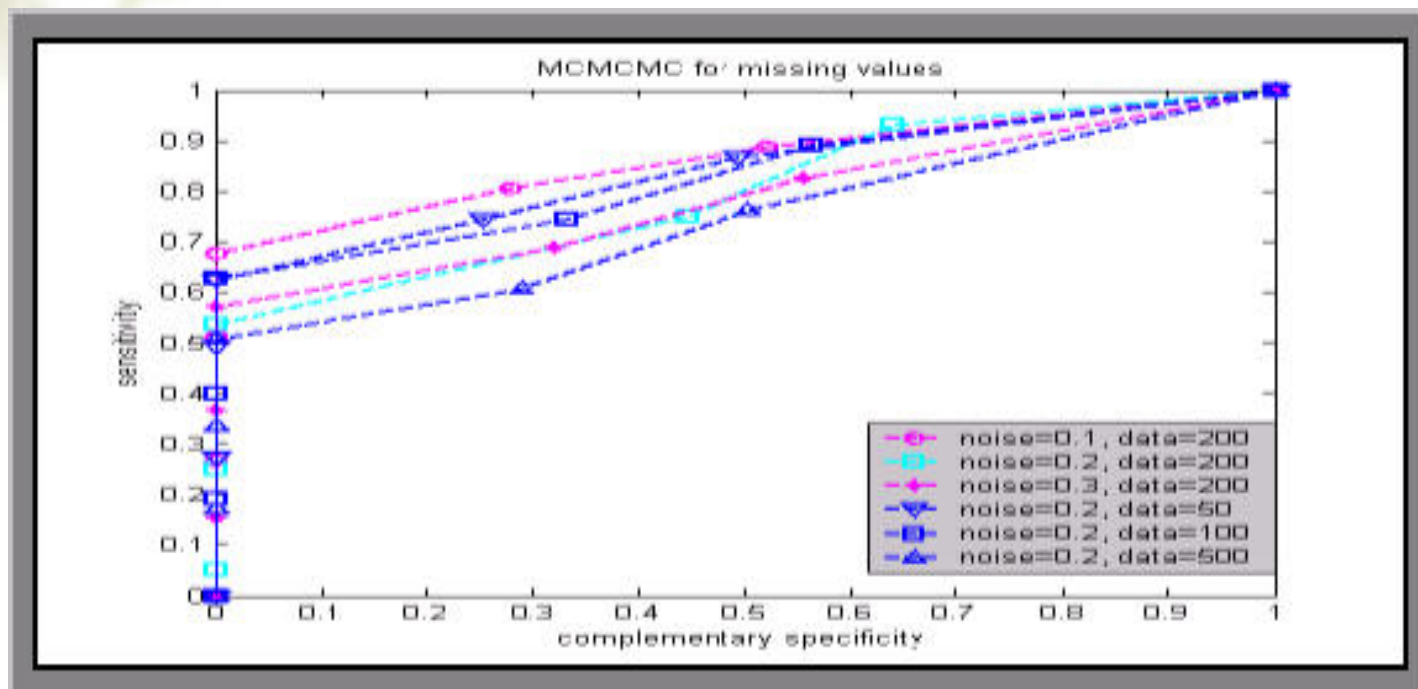


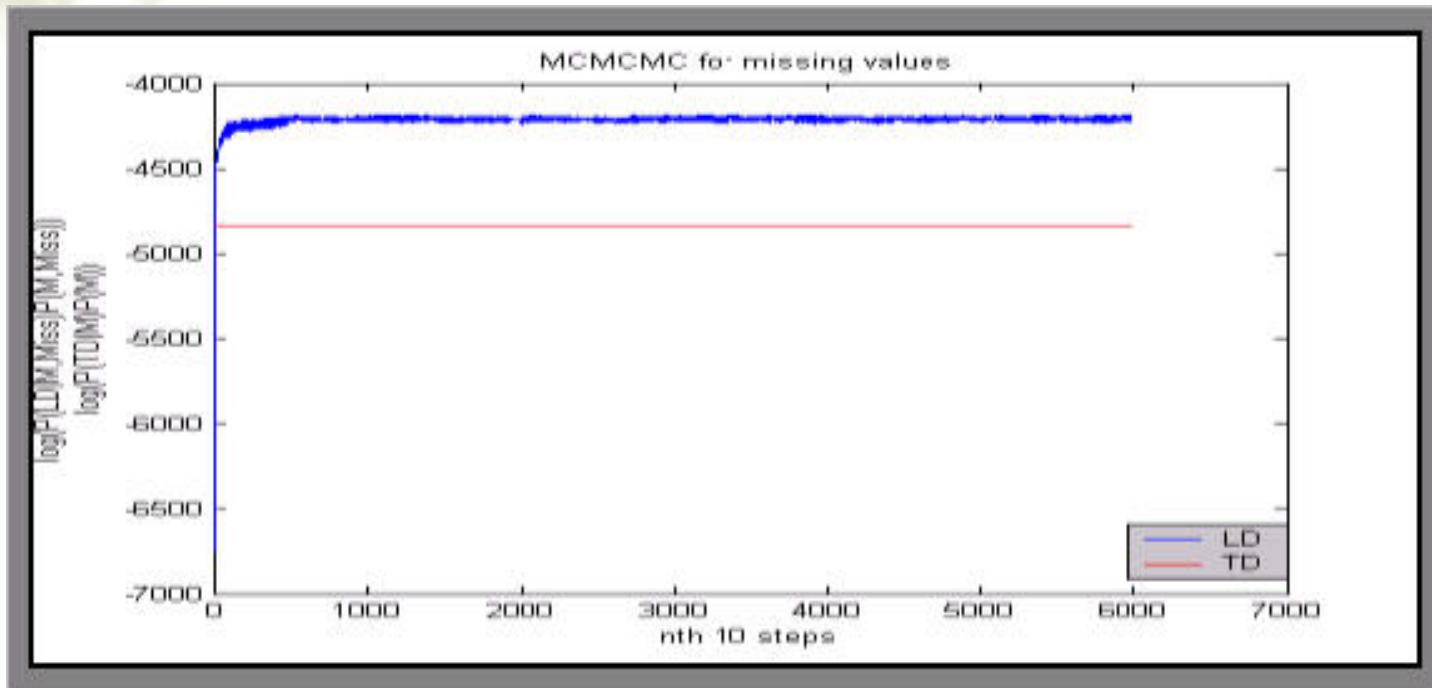
# Result Evaluation (missing values)

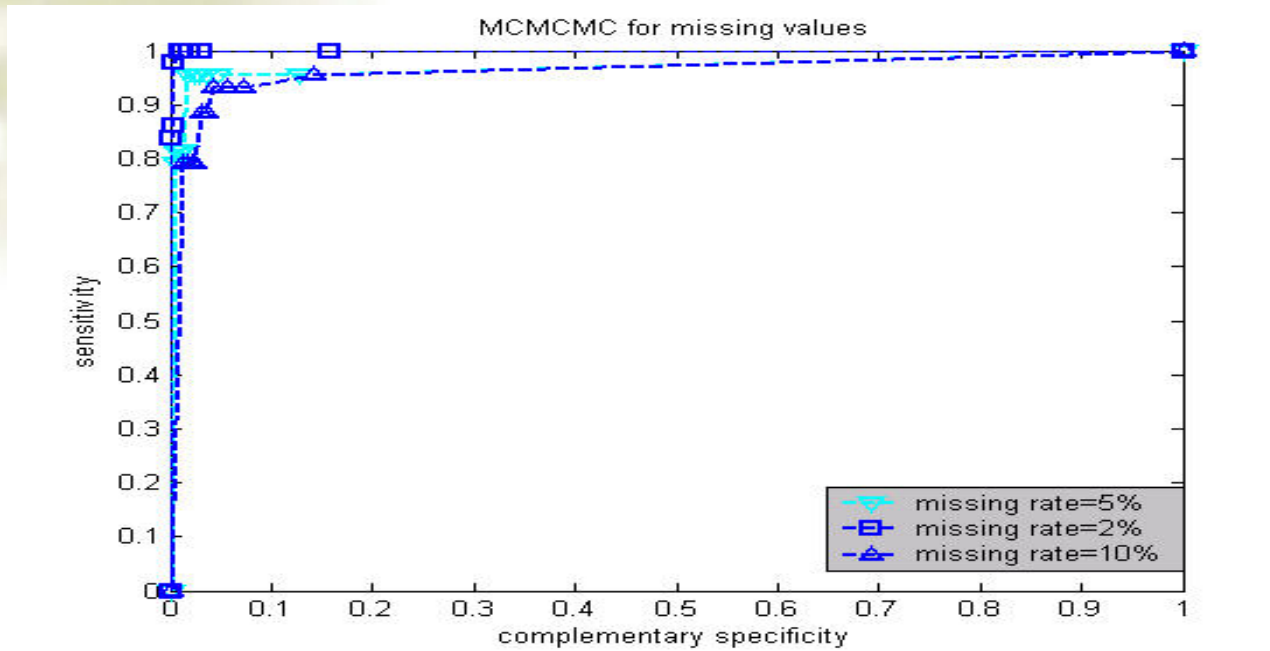
## ❖ Model Genetic Network



- ❖ Before burn in(30000 burn in, 30000 iterations)
- ❖ After burn in 40000 iterations
- ❖ Temp=[1,1,3,9,12]

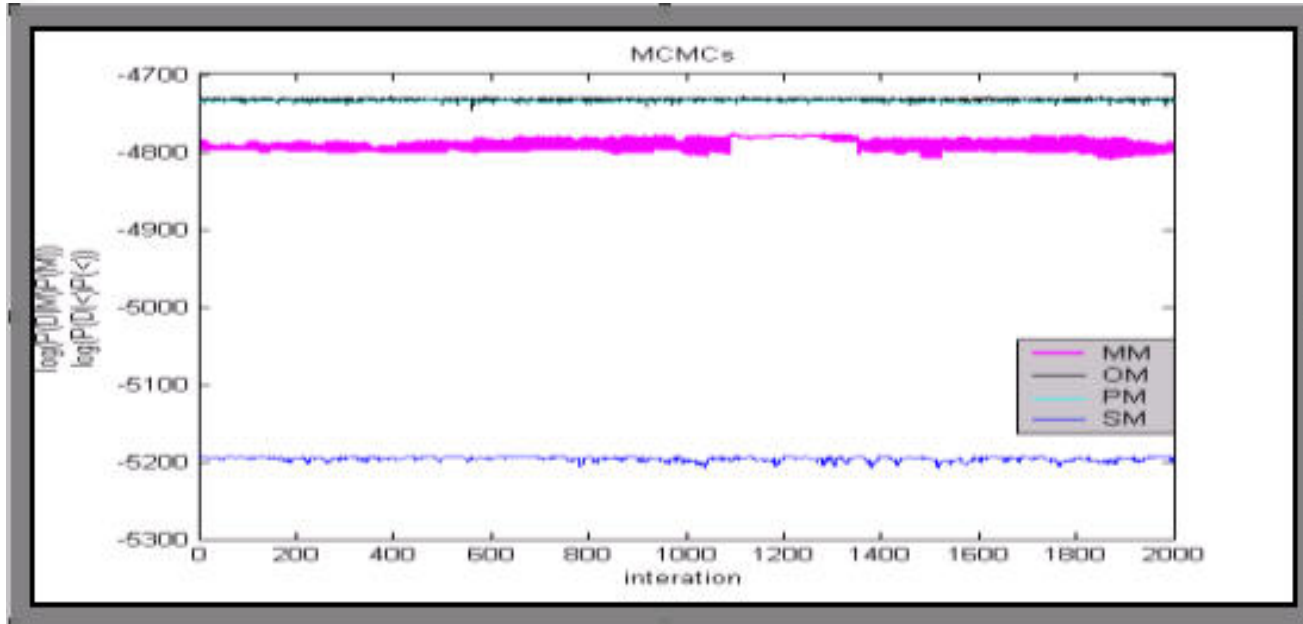




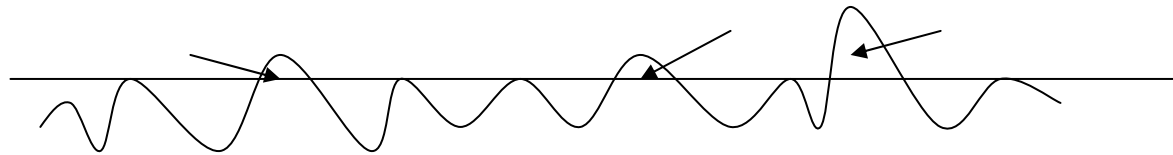


- ❖ The ROC curve for noise=0.2 data=200 with different missing rate
- ❖ Temp = [1, 1, 3, 9, 12]
- ❖ Use 30000 burn in and 30000 iterations.  
Every 10 steps keep one sample. (before burn in algorithm)

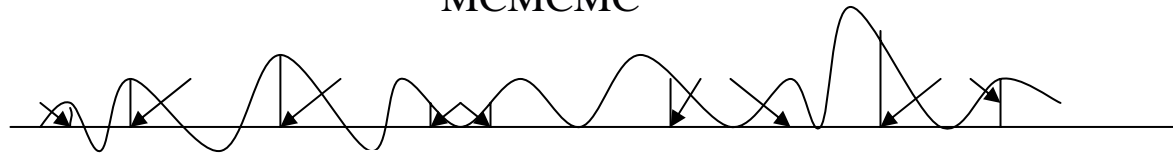
## ❖ B cell Lymphoma data



# Summary



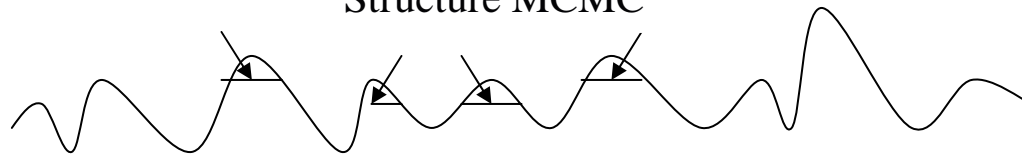
MCMCMC



Order MCMC



Structure MCMC



Population MCMC

## ❖ Problems with MCMCMC

